1.How many seconds are in an hour? Use the interactive interpreter as a calculator and multiply the number of seconds in a minute (60) by the number of minutes in an hour (also 60).

hour = int(input("Enter an hours"))\

seconds\_per\_hour = hour \* 60 \* 60

print(seconds\_per\_hour)

2. Assign the result from the previous task (seconds in an hour) to a variable called seconds\_per\_hour.

3. How many seconds do you think there are in a day? Make use of the variables seconds per hour and minutes per hour.

seconds\_per\_day = 24 \* 60 \* 60

minutes\_per\_day = 24 \* 60

print(seconds\_per\_day)

print(minutes\_per\_day)

4. Calculate seconds per day again, but this time save the result in a variable called seconds\_per\_day

seconds\_per\_day = 24 \* 60 \* 60

minutes\_per\_day = 24 \* 60

print(seconds\_per\_day)

print(minutes\_per\_day)

5. Divide seconds\_per\_day by seconds\_per\_hour. Use floating-point (/) division.

print(float(seconds\_per\_day/seconds\_per\_hour))

6. Divide seconds\_per\_day by seconds\_per\_hour, using integer (//) division. Did this number agree with the floating-point value from the previous question, aside from the final .0?

print(int(seconds\_per\_day/seconds\_per\_hour))

7. Write a generator, genPrimes, that returns the sequence of prime numbers on successive calls to its next() method: 2, 3, 5, 7, 11, ...

primes = [2,3]

no = primes[-1]

def genPrimes():

     global no

     no = no + 2

     # print(f"next no:{no}")

     val = list(filter(lambda x: no % x != 0, primes))

     # print(f"lambda val - {list(val)}")

     # primes.append(no)

     if (len(primes)==len(val)):

         primes.append(no)

         print(f"primes- {list(primes)}")

     else:

          genPrimes()